

#3 4601  
Brüderly



## Bescheinigung

Die Philips Corporate Intellectual Property GmbH in Aachen /Deutschland hat eine Patentanmeldung unter der Bezeichnung

"Erkennung einer in buchstabierter Form vorliegenden Sprach-  
äußerungseingabe"

am 17. September 1999 beim Deutschen Patent- und Markenamt eingereicht.

Der Sitz der Anmelderin wurde berichtet in: Hamburg/Deutschland.

Die angehefteten Stücke sind eine richtige und genaue Wiedergabe der ursprünglichen Unterlagen dieser Patentanmeldung.

Die Anmeldung hat im Deutschen Patent- und Markenamt vorläufig die Symbole G 10 L und H 04 M der Internationalen Patentklassifikation erhalten.

München, den 9. März 2000

Deutsches Patent- und Markenamt

Der Präsident

Im Auftrag

Aktenzeichen: 199 44 608.3

Jerofsky



## ZUSAMMENFASSUNG

### Erkennung einer in buchstabierter Form vorliegenden Sprachäußerungseingabe

Die Erfindung betrifft ein Verfahren zur Erkennung einer in buchstabierter Form vorliegenden Sprachäußerungseingabe (s) mit einer ersten Verarbeitungsstufe, in der mittels  
5 einer auf Hidden Markov Modellen basierenden Buchstaben-Spracherkennungseinheit (2) eine korrespondierende Buchstabenfolge (r) geschätzt wird, und mit einer zweiten Verarbeitungsstufe (3), in der unter Verwendung eines statistischen Buchstabenfolgmodells (4) und eines statistischen Modells (5) für die Spracherkennungseinheit (2) das von der ersten Verarbeitungsstufe gelieferte Schätzergebnis (r) nachverarbeitet wird, wobei bei der  
10 Nachverarbeitung die Methode der dynamische Programmierung eingesetzt wird.

Um robuste und effiziente Spracherkennungsprozeduren für die Anwendung von Sprachsignalen zur Systemsteuerung, wird vorgeschlagen, daß die der dynamischen Programmierung zugrundeliegende Gitterstruktur, deren Knotenpunkte zur Zuordnung zu akkumulierten Wahrscheinlichkeitswerten vorgesehen sind, in eine Baumstruktur umgesetzt wird  
15 und daß bei der Suche nach einem optimalen Baumpfad der A\*-Algorithmus eingesetzt wird.

Es wird auch ein Verfahren vorgeschlagen, beim dem im Rahmen einer Sprachsteuerung  
20 eine Eingabe eines als Steuersignal dienenden ganzen Wortes und eine Eingabe wenigstens eines Teils dieses Wortes in buchstabierter Form vorgesehen ist, wobei das Ergebnis der Buchstaben-Spracherkennung im Rahmen der Wort-Spracherkennung verwendet wird.

Fig. 4

## BESCHREIBUNG

Erkennung einer in buchstabierter Form vorliegenden Sprachäußerungseingabe

Die Erfindung betrifft ein Verfahren zur Erkennung einer in buchstabierter Form vorliegenden Sprachäußerungseingabe mit einer ersten Verarbeitungsstufe, in der mittels einer  
5 auf Hidden Markov Modellen basierenden Buchstaben-Spracherkennungseinheit eine korrespondierende Buchstabenfolge geschätzt wird, und mit einer zweiten Verarbeitungsstufe, in der unter Verwendung eines aus dem Vokabular abgeleiteten statistischen Buchstabenfolgemodells und eines statistischen Modells für die Spracherkennungseinheit das von der ersten Verarbeitungsstufe gelieferte Schätzergebnis nachverarbeitet wird, wobei  
10 bei der Nachverarbeitung die Methode der dynamischen Programmierung eingesetzt wird.

Ein solches Verfahren ist beispielsweise aus der US 5,799,065 im Rahmen der automatischen Herstellung von Telefonverbindungen durch Spracheingaben bekannt. Ein Anrufer gibt dabei nach entsprechender Aufforderung den Namen des gewünschten anzurufenden  
15 anderen Teilnehmers in kontinuierlich buchstabierter Form ein. Die Eingabe wird einer auf HMM (Hidden Markov Model) verwendenden Spracherkennungseinheit weiterverarbeitet, wobei auch eine auf Buchstaben-n-Grammen beruhende Buchstabengrammatik zum Einsatz kommt. Es werden N beste Worthypothesen ermittelt, die nach der Methode des dynamischen Programmierens (DP) weiterverarbeitet werden, wobei die ermittelten  
20 Hypothesen mit dem Inhalt eines Namenslexikons verglichen werden. Die von der DP-Einheit gelieferten N besten Worthypothesen werden als dynamische Grammatik verwendet, die von einer weiteren Spracherkennungseinheit verwendet wird, die aus den von der DP-Einheit gelieferten Worthypothesen eine als - dem eingegebenen Namen entsprechenden - Erkennungsergebnis auswählt.

25 Bei Navigationssystemen für Landkraftfahrzeuge ist es ebenfalls bekannt, Eingaben mittels Sprachäußerungen zu tätigen. Auf diese Weise werden beispielsweise Ortsnamen als Zielorte eingegeben. Um die Zuverlässigkeit der Spracherkennung zu verbessern, ist neben einer Wort-Spracherkennung, bei der eine Eingabe von natürlich gesprochenen Worten  
30 vorgesehen, außerdem eine Buchstaben-Spracherkennung vorgesehen, die zur Erkennung buchstabierter Spracheingaben dient.

Der Erfindung liegt nun die Aufgabe zugrunde, robuste und effiziente Spracherkennungsprozeduren für die Anwendung von Sprachsignalen zur Systemsteuerung unter Anwendung einer Buchstaben-Spracherkennung anzugeben.

5

Die Aufgabe wird dadurch gelöst, daß die der dynamischen Programmierung zugrundeliegende Gitterstruktur, deren Knotenpunkte zur Zuordnung zu akkumulierten Wahrscheinlichkeitswerten vorgesehen sind, in eine Baumstruktur umgesetzt wird und daß bei der Suche nach einem optimalen Baumpfad der A\*-Algorithmus eingesetzt wird. Dieser Ansatz führt zu einer beschleunigten Buchstabenspracherkennung mit verringertem Speicherplatzbedarf.

10

In einer Ausgestaltung ist vorgesehen, daß suboptimale Baumpfade entsprechend N bester Schätzungen für die Sprachäußerungseingabe mit  $N > 1$  ermittelt werden. Damit stehen Erkennungsalternativen für eine Weiterverarbeitung zur Verfügung, so daß ein Fehler bei der Ermittlung des optimalen Baumpfades in nachfolgenden Verarbeitungsschritten unter Zuhilfenahme der suboptimalen Erkennungsergebnisse leichter korrigierbar ist.

15

Eine weitere Ersparnis an Rechenzeit wird dadurch erreicht, daß bei der Suche nach einem optimalen Baumpfad diejenigen Baumpfade, die schon zu Beginn der Suche eine gegenüber anderen Baumpfaden eine kleine Wahrscheinlichkeit aufweisen, vorrangig nicht mehr weiterverfolgt werden.

20

Es wird weiterhin vorgeschlagen, daß die erste Verarbeitungsstufe mittels eines ersten ICs und die zweite Verarbeitungsstufe mittels eines zweiten ICs durchgeführt wird. Der erste IC (integrierte Schaltkreis) ist vorzugsweise ein speziell für Spracherkennungsprozeduren programmierte digitaler Signalprozessor. Der zweite IC kann insbesondere ein Controller-Baustein sein, der auch zur Realisierung anderer Systemfunktionen eingesetzt wird.

25

Die Erfindung betrifft auch ein Verfahren zur Systemsteuerung mittels Sprachsignalen, bei dem

30

- eine Eingabe eines als Steuersignal dienenden ganzen Wortes und eine Eingabe wenigstens eines Teils dieses Wortes in buchstabierter Form vorgesehen ist,
- eine Wort-Spracherkennung zur Erkennung des eingegebenen ganzen Wortes vorgesehen ist,
- 5 - eine Buchstaben-Spracherkennung, insbesondere wie oben beschrieben, zur Erkennung des eingegebenen buchstabierten Teils des ganzen Wortes vorgesehen ist und
- bei dem mit Hilfe des Erkennungsergebnisses der Buchstaben-Spracherkennung eine Einschränkung eines der Wort-Spracherkennung zugeordneten Vokabulars durchgeführt wird.

10

Ein solches Verfahren führt auch für schwierige Rahmenbedingungen wie beispielsweise einem hohen Störgeräuschpegel in Kraftfahrzeugen oder undeutlicher Sprechweise eines Benutzers zu einer zuverlässigen Sprachsteuerung.

- 15 Die Erfindung betrifft auch ein sprachgesteuertes elektrisches Gerät, insbesondere ein Navigationssystem für Landkraftfahrzeuge, mit Komponenten zur Durchführung eines der oben beschriebenen Verfahren.

Ausführungsbeispiele der Erfindung werden nachfolgend unter anderem anhand der

- 20 Zeichnungen näher erläutert. Es zeigen:

Fig. 1 eine Baumstruktur zur Erläuterung zum statistischen Modell einer Buchstabenfolge,

Fig. 2 ein Beispiel für einen Gitterpfad,

- 25 Fig. 3 eine Baumstruktur, deren Baumknoten Spalten eines DP-Gitters entsprechen,

Fig. 4 ein Blockschaltbild eines Systems zur Erkennung buchstabierter Sprachäußerungen und

Fig. 5 ein Blockschaltbild eines Systems mit Sprachsteuerung durch Eingabe von Worten und buchstabierter sprachlicher Äußerungen.

Eine bevorzugte Anwendung der Erfindung ist ein Navigationssystem für Kraftfahrzeuge mit einer Sprachsteuerung. Die automatische Spracherkennung für die Sprachsteuerung ist hier schwierig, da das zu erkennende Vokabular (z. B. einige zehntausend Städtenamen) umfangreich ist und die akustischen Bedingungen in Kraftfahrzeugen aufgrund zahlreicher auftretender Störgeräusche als ungünstig beurteilt werden müssen. Weiterhin ist davon auszugehen, daß die verfügbare Hardware in Navigationssystemen in Anbetracht der Komplexität von Spracherkennungsprozeduren nur eine sehr beschränkte Verarbeitungskapazität und einen relativ kleinen Arbeitsspeicher aufweist. Die Erfindung ist allerdings nicht auf die Anwendung in Navigationssystemen für Kraftfahrzeuge beschränkt, sondern in allen Geräte mit Sprachsteuerung und ähnlich gelagerten Randbedingungen geeignet.

Beim vorliegenden Navigationssystem wird ein Benutzer im Spracherkennungsmodus zur Spracheingabe aufgefordert, z. B. zur Eingabe eines Städtenamens, und zwar jeweils sowohl zur Eingabe durch Sprechen eines ganzen Wortes als auch zusätzlich zur Eingabe durch (kontinuierliches) Buchstabieren wenigstens eines Teils des eingegebenen Wortes. Es wird in zwei ersten Verarbeitungsstufen sowohl eine auf dem vorgegebenen Vokabular basierende Wort-Spracherkennung als auch eine Buchstaben-Spracherkennung durchgeführt. Bei der Buchstaben-Spracherkennung wird dem Benutzer die Anzahl einzugebender Buchstaben pro Wort nicht vorgegeben. Mit dem Spracherkennungsergebnis bezüglich der eingegebenen Einzelbuchstaben können die Worte des vorgegebenen Vokabulars ermittelt werden, die als Wort-Spracherkennungsergebnis in Frage kommen. Basierend auf dem sich hieraus ergebenden eingeschränkten Vokabular wird in einer weiteren Verarbeitungsstufe für das eingegebene Wort erneut eine Wort-Spracherkennung durchgeführt.

Im folgenden soll die Buchstaben-Spracherkennung näher erläutert werden. Bei dieser ist regelmäßig, insbesondere in Umgebungen mit erheblichen Störgeräuschen wie innerhalb von Kraftfahrzeugen, mit hohen Fehlerraten zu rechnen. Einer Verbesserung dieser Fehler-rate durch Berücksichtigung des Vokabulars bei der akustischen Suche des Buchstaben-Spracherkenners stößt auf das Problem, daß übliche Spracherkenner-ICs nicht genügend Speicher zur Speicherung der aus einem großen Vokabular resultierenden Datenmengen enthalten. Bei dem vorliegenden Navigationssystem wird aus diesem Grund die Buchstaben-Spracherkennung in zwei voneinander unabhängigen Verarbeitungsstufen

durchgeführt. In der ersten Verarbeitungsstufe werden die eingegebenen Buchstaben mittels eines üblichen Buchstaben-Spracherkenners ohne Berücksichtigung eines Vokabulars durchgeführt. Diese Verarbeitungsstufe wird mittels eines speziell hierfür ausgelegten und programmierten Spracherkenner-ICs durchgeführt. In der zweiten Verarbeitungsstufe wird eine Nachverarbeitung durchgeführt. Diese wird mittels des Controllers durchgeführt, der zur Umsetzung der anderen Systemfunktionen (d.h. hier der speziellen Navigationsfunktionen) dient und der auf ausreichend Speicherplatz zugreifen kann.

Für die Nachverarbeitung stehen zusätzliche Informationen bezüglich verschiedener möglicher Buchstabenfolgen zur Verfügung, insbesondere - wie im vorliegenden Ausführungsbeispiel - eine Liste mit zulässigen Buchstabenfolgen, d.h. Buchstabenfolgen, mit denen jeweils mindestens ein Wort des Vokabulars beginnt, und statistische Informationen bezüglich solcher Buchstabenfolgen, z.B. bedingte Wahrscheinlichkeiten (wie beispielsweise die Wahrscheinlichkeit, daß beim dritten Buchstaben C eines Wortes die beiden ersten Buchstaben jeweils ein A waren). Als weitere die Fehlerrate reduzierende statistische Informationen kommen auch Wahrscheinlichkeiten für Verwechslungen zwischen zwei Buchstaben (N und M sind z. B. einander ähnlich und haben deshalb eine hohe Verwechslungswahrscheinlichkeit) oder Wahrscheinlichkeiten bzgl. eines versehentlichen Einfügens oder Auslassens eines Buchstabens in Betracht.

20

Das der Nachverarbeitung zugrundeliegende Problem läßt sich wie folgt formulieren:

Gegeben sind:

- ein statistisches Modell des Buchstaben-Spracherkenners (d.h. Wahrscheinlichkeiten von Erkennungsfehlern);
- ein statistisches Modell der gesprochenen Buchstabenfolge und
- eine Folge erkannter Buchstaben.

25

Gesucht ist:

- 30 Die Buchstabenfolge mit der größten Wahrscheinlichkeit, die gesprochene Buchstabenfolge zu sein.

Im folgenden wird  $\Sigma$  als Symbol für eine Buchstabenmenge bezeichnet.

Eine gesprochene (und in den Spracherkenner eingegebene) Buchstabenfolge  $\underline{s}$  der Länge  $n$  (mit Buchstaben  $s_i$ ) und eine erkannte Buchstabenfolge  $\underline{r}$  der Länge  $m$  (mit Buchstaben  $r_i$ )

5 werden beschrieben durch:

$$\underline{s} = (s_1, s_2, \dots, s_n) \text{ mit } s_i \in \Sigma$$

$$\underline{r} = (r_1, r_2, \dots, r_m) \text{ mit } r_i \in \Sigma$$

Die Buchstabenfolgen sind zur Unterscheidung gegenüber Einzelbuchstaben unterstrichen  
10 dargestellt. Unterschiedliche Längen  $n$  und  $m$  können sich dadurch ergeben, daß der verwendete Spracherkenner fehlerhaft Buchstaben ins Erkennungsergebnis einfügt oder auch fehlerhaft Buchstaben wegläßt.

Gesucht ist nun die Buchstabenfolge  $\underline{s}$ , bei der bei gegebener Buchstabenfolge  $\underline{r}$  die Wahr-  
15 scheinlichkeit

$$P(\underline{s} | \underline{r}) = \frac{P(\underline{r} | \underline{s}) P(\underline{s})}{P(\underline{r})}$$

maximal ist. Da das Wahrscheinlichkeitsmaximum von  $P(\underline{r})$  unabhängig ist, ist nach der Buchstabenfolge  $\underline{s}$  zu suchen, die den Ausdruck

20

$$P(\underline{r} | \underline{s}) P(\underline{s})$$

maximiert. Der Wahrscheinlichkeitsterm  $P(\underline{r} | \underline{s})$  beschreibt die Spracherkennereigenschaften (durch die Wahrscheinlichkeit für eine Folge erkannter Buchstaben  $\underline{r}$  gegeben eine  
25 Folge gesprochener Buchstaben  $\underline{s}$ ), der Wahrscheinlichkeitsterm  $P(\underline{s})$  dagegen die Auftretenswahrscheinlichkeiten von gesprochenen Buchstabenfolgen  $\underline{s}$  (entsprechend einem Sprachmodell, das berücksichtigt, daß nicht alle Buchstabenkombinationen gleich wahrscheinlich sind).



Zur Berechnung des Maximums des Ausdrucks  $P(\underline{r} \mid \underline{s}) P(\underline{s})$  soll ein effizienter Algorithmus angegeben werden. Hierzu werden vereinfachende Annahmen bezüglich der beiden Wahrscheinlichkeitsfunktionen  $P(\underline{r} \mid \underline{s})$  und  $P(\underline{s})$  gemacht, um so geeignete statistische Modelle für den Spracherkenner und die gesprochene Buchstabenfolge zu erhalten. Im folgenden  
 5 wird das statistische Modell für  $P(\underline{r} \mid \underline{s})$  mit  $P_R(\underline{r} \mid \underline{s})$  und das statistische Modell für  $P(\underline{s})$  mit  $P_S(\underline{s})$  bezeichnet.

Als (aus dem vorgegebenen Vokabular abgeleitetes) statistisches Modell für die gesprochene Buchstabenfolge wird nun der Ausdruck

10

$$P_S(s_{i+1} \mid s_1, \dots, s_i)$$

angesetzt, der die Wahrscheinlichkeit dafür angibt, daß auf eine Folge von  $i$  gesprochenen Buchstaben  $s_1, \dots, s_i$  als nächster gesprochener Buchstabe  $s_{i+1}$  folgt. Die Wahrscheinlichkeit,  
 15 daß sprachliche Äußerung nach den Buchstaben  $s_1, \dots, s_i$  endet, ist gegeben durch

$$P_S(\$ \mid s_1, \dots, s_i) = 1 - \sum_{s_{i+1} \in \Sigma} P_S(s_{i+1} \mid s_1, \dots, s_i) ,$$

wobei \$ das Ende einer Buchstabenfolge bezeichnet. Derartige Wahrscheinlichkeiten können leicht aus einem gegebenen Vokabular und a priori Wahrscheinlichkeiten für die  
 20 Wörter des Vokabulars geschätzt werden. Dementsprechend läßt sich die Wahrscheinlichkeit für eine Folge gesprochener Buchstaben  $\underline{s} = (s_1, s_2, \dots, s_n)$  angeben durch

$$P_S(\underline{s}) = P_S(s_1 \mid \#) P_S(s_2 \mid s_1) \dots P_S(s_n \mid s_1, \dots, s_{n-1}) P_S(\$ \mid s_1, \dots, s_n) ,$$

25 wobei das Zeichen # den Beginn einer Buchstabenfolge bezeichnet. Weiterhin wird ein begrenztes Vokabular  $V$  mit

$$V = \{ \underline{s} \mid P_S(\underline{s}) \neq 0 \}$$

- vorausgesetzt. Für den Fall, daß eine Buchstabenfolge  $s$  Element des Vokabulars  $V$  ist, ist auch jedes Präfix von  $s$  (d.h. eine Folge von einem oder mehreren aufeinanderfolgenden Buchstaben, mit denen die Buchstabenfolge  $s$  beginnt) Element des Vokabulars  $V$ . Dadurch kann der Benutzer eine beliebig lange Anfangsbuchstabenkette des zu buchstabierenden Wortes sprechen und muß das Wort nicht vollständig buchstabieren. Durch entsprechende Wahl von  $P_s$  kann a-priori-Wissen über die Wahrscheinlichkeit ausgenutzt werden, wie viele Buchstaben ein Benutzer voraussichtlich bei der Eingabe im Buchstabiermodus sprechen wird.
- 10 Die verschiedenen Wahrscheinlichkeiten  $P_s$  eines Vokabulars  $V$  lassen sich mit Hilfe einer Baumstruktur auf einfache Weise darstellen. Dabei wird einer Baumkante jeweils ein Buchstabe und der zugehörige Wahrscheinlichkeitswert zugeordnet. Jeder gesprochenen Buchstabenfolge entspricht dann ein Baumknoten, wobei sich die Wahrscheinlichkeit der Buchstabenfolge aus dem Produkt derjenigen Wahrscheinlichkeiten ergibt, die den von der
- 15 Baumwurzel bis zum betreffenden Baumknoten führenden Baumkanten zugeordnet sind.

- Ein Beispiel für eine derartige Baumstruktur zeigt Fig. 1. Hier sind zur Bildung des Vokabulars vereinfacht als mögliche Buchstaben A, B, C, D und E vorausgesetzt, die jeweils zusammen mit der zugehörigen Auftrittswahrscheinlichkeit einer Baumkante zugeordnet
- 20 sind. Dementsprechend ergeben sich für die Buchstabenfolgen AB, AC und DE die Wahrscheinlichkeitswerte  $P_s(AB)=0,18$ ,  $P_s(AC)=0,06$  und  $P_s(DE)=0,56$  als Produkt der den einzelnen Buchstaben der Buchstabenfolgen jeweils zugeordneten Wahrscheinlichkeitswerten. Unter der Bedingung, daß die Wahrscheinlichkeit des Erreichens eines Endes  $s$  einer Buchstabenfolge schon vor dem Durchlauf eines ganzen Baumpfades mit  $P_s(s)=0,2$
- 25 angesetzt wird, ergeben sich weiterhin die Wahrscheinlichkeitswerte  $P_s(A)=0,06$  und  $P_s(D)=0,14$  durch Multiplikation von  $P_s(s)$  mit den den Buchstaben A und D bzw. den zugehörigen Baumkanten zugeordneten Wahrscheinlichkeiten. Die Summe der Wahrscheinlichkeitswerte  $P_s$  ergibt sich zu Eins.
- 30 Um ein einfaches statistisches Modell für den Buchstaben-Spracherkenner (genauer gesagt: für die Erkennungsfehler des Buchstaben-Spracherkenners) zu erhalten, wird vorausgesetzt, daß die gesprochenen Buchstaben unkorreliert sind und lediglich die

Korrelationen zwischen einem erkannten und einem gesprochenen Buchstaben berücksichtigt werden. Das statistische Modell für den Buchstaben-Spracherkenner liefert die Wahrscheinlichkeit, daß ein Buchstabe  $r$  erkannt und ein Buchstabe  $s$  gesprochen wurde (mit  $r, s \in \Sigma$ ). Weiterhin werden bei diesem Modell Wahrscheinlichkeiten für Einfügungen von

5 Buchstaben  $r$  ohne korrespondierenden gesprochenen Buchstaben  $s$  und Wahrscheinlichkeiten für Löschungen von Buchstaben (kein erkannter Buchstabe  $r$  für den Fall eines gesprochenen Buchstabens  $s$ ) angesetzt. Zur Beschreibung dieser Fälle wird ein virtueller Buchstabe  $\varepsilon \notin \Sigma$  eingeführt, der sowohl zur Bezeichnung eines nicht gesprochenen als auch zur Bezeichnung eines nicht erkannten Buchstabens dient. Demgemäß ergibt sich für

10 das statistische Modell des Buchstaben-Spracherkenners zu

$$P_R(r, s) \text{ mit } r, s \in \Sigma \cup \{\varepsilon\}.$$

Diese Verbund-Wahrscheinlichkeiten werden als Elemente einer Matrix ("confusion

15 matrix") aufgefaßt, wobei die verschiedenen Buchstaben  $r$  und  $s$  den einzelnen Zeilen bzw. Spalten der Matrix bezeichnen. Ausgehend von dieser als gegeben vorausgesetzten und in gespeicherter Form vorliegenden Matrix werden bedingte Wahrscheinlichkeiten  $P(\underline{r} \mid \underline{s})$  für eine erkannte Buchstabenfolge  $\underline{r}$  und eine gesprochene Buchstabenfolge  $\underline{s}$  berechnet, was nachstehend noch näher erläutert wird.

20

Um mögliche zeitliche Verläufe von Zuordnungen von Buchstabenfolgen  $\underline{r}$  und  $\underline{s}$  darzustellen, wird ein zweidimensionales Gitter benutzt, das in vertikaler Richtung  $m+1$  Punkte und in horizontaler Richtung  $n+1$  Punkte aufweist, die jeweils mit einem  $r_i$  und einem  $s_i$  bezeichnet sind. Die 0-te Zeile und 0-te Spalte bleiben dabei unbezeichnet. Der

25 zeitliche Verlauf einer Zuordnung einer bestimmten Buchstabenfolge  $\underline{r}$  zu einer bestimmten Buchstabenfolge  $\underline{s}$  entspricht einem Pfad durch ein solches Gitter, der durch eine Folge  $\pi$  von Koordinatenpaaren

$$\pi = (\sigma_1, \rho_1), (\sigma_2, \rho_2), \dots, (\sigma_k, \rho_k)$$

30 mit

$$\sigma_1 = \rho_1 = 0 ;$$

$$(\sigma_{i+1}, \rho_{i+1}) \in \{(\sigma_i, \rho_i + 1), (\sigma_i + 1, \rho_{i+1}), (\sigma_i + 1, \rho_i + 1)\} ;$$

$$\sigma_i \leq n, \rho_i \leq m .$$

- 5 Ein Pfadsegment  $(\sigma_{i-1}, \rho_{i-1}) \rightarrow (\sigma_i, \rho_i)$ , bei dem sowohl die  $\sigma$ -Koordinate als auch die  $\rho$ -Koordinate inkrementiert worden sind, impliziert, daß ein Buchstabe  $s_{\sigma_i}$  gesprochen und ein Buchstabe  $r_{\rho_i}$  erkannt wurde. Ist bei einem Pfadsegment jedoch die  $\sigma$ -Koordinate konstant, dann wurde zwar ein Buchstabe  $r_{\rho_i}$  erkannt, aber kein Buchstabe gesprochen, was dem fehlerhaften Einfügen eines Buchstaben durch den Buchstaben-Spracherkenner
- 10 entspricht. Ist die  $\rho$ -Koordinate konstant, wurde zwar ein Buchstabe  $s_{\sigma_i}$  gesprochen, jedoch kein vom Spracherkenner kein korrespondierender Buchstabe erkannt (Löschfehler).

Fig. 2 zeigt beispielhaft einen Ausschnitt aus einer derartigen Gitterstruktur mit einem eingezeichneten Pfad  $\pi$ . Es wurden drei Buchstaben  $s_1, s_2$  und  $s_3$  gesprochen und zwei

15 Buchstaben  $r_1$  und  $r_2$  erkannt. Der Buchstabe  $r_1$  wurde als Buchstabe  $s_1$  erkannt. Der Buchstabe  $s_2$  wurde nicht erkannt (d.h. gelöscht). Der Buchstabe  $s_3$  wurde schließlich als Buchstabe  $r_2$  erkannt.

Allgemein ergibt sich die Wahrscheinlichkeit  $P_R$  für eine gesprochene Buchstabenfolge  $\underline{s}$ ,

20 eine erkannte Buchstabenfolge  $\underline{r}$  und einen Gitterpfad  $\pi$  gemäß

$$P_R(\underline{r} | \underline{s}) = \sum_{\pi} \prod_{i=1}^{|\pi|} \begin{cases} P_R(r_{\rho_i} | s_{\sigma_i}) & \text{falls } \rho_i \neq \rho_{i-1} \quad \text{und } \sigma_i \neq \sigma_{i-1} \\ P_R(\varepsilon | s_{\sigma_i}) & \text{falls } \rho_i = \rho_{i-1} \quad \text{und } \sigma_i \neq \sigma_{i-1} \\ P_R(r_{\rho_i}, \varepsilon) & \text{falls } \rho_i \neq \rho_{i-1} \quad \text{und } \sigma_i = \sigma_{i-1} \end{cases}$$

Dabei wird in der dritten Zeile für den Fall, daß zwar ein Buchstabe erkannt wurde, aber kein entsprechender gesprochener Buchstabe vorliegt, eine Verbundwahrscheinlichkeit anstelle einer bedingten Wahrscheinlichkeit (wie in den beiden oberen Zeilen) für  $P_R$

25 angesetzt.

Zusammenfassend stellt sich das der Buchstaben-Spracherkennung zugrundeliegende Problem somit so dar, daß diejenige gesprochene Buchstabenfolge  $\underline{s} \in V$  zu ermitteln ist, die bei einer gegebenen erkannten Buchstabenfolge  $\underline{r}$  die Funktion  $f(\underline{s})$  mit

5 
$$f(\underline{s}) = P_R(\underline{r} | \underline{s}) P_S(\underline{s})$$

maximiert.

Eine Verbesserung der Buchstaben-Spracherkennung ergibt sich bei der Verwendung eines  
10 Buchstaben-Spracherkenners, der nicht nur einzelne Buchstaben als Hypothesen für jeweils einen gesprochenen Buchstaben ausgibt, sondern eine Liste  $N$  bester Buchstabenhypothesen ( $N > 1$ ), die jeweils mit einem Wahrscheinlichkeitswert gewichtet sind. Diese erweiterte Ergebnisinformation kann völlig analog zu den obigen Ausführungen verarbeitet werden (also Verarbeitung ebenfalls ausgehend von einer Matrix und einer Gitterstruktur), was zu  
15 einer verbesserten Erkennungsfehlerrate führt.

Im folgenden soll die Nachverarbeitung beschreiben werden, durch die das oben genannte Problem der Maximierung von  $f(\underline{s})$  gelöst wird.

20 Im folgenden sei  $\Sigma$  ein vorgegebenes Alphabet,  $V \subseteq \Sigma^*$  ein begrenztes Vokabular mit  $\Sigma^*$  als Menge möglicher Buchstabenketten, so daß im Falle einer gesprochenen Buchstabenfolge  $\underline{s} \in V$  auch jedes Präfix der Buchstabenfolge  $\underline{s}$  Element von  $V$  ist.  $P_R$ ,  $P_S$  und  $f(\underline{s})$  sollen wie oben definiert sein. Weiterhin sei  $\underline{r} \in \Sigma^*$  eine beliebige, aber feste Folge erkannter Buchstaben.

25

Eine (direkte) Möglichkeit zur Bestimmung der Folge  $\underline{s}$  mit der größten Wahrscheinlichkeit ist die Berechnung aller Werte  $f(\underline{s})$  für alle  $\underline{s} \in V$ , wobei die gesuchte Folge  $\underline{s}$  diejenige ist, bei der  $f(\underline{s})$  maximal ist. Zur Bewertung von  $f(\underline{s})$  bietet sich eine leicht modifizierte Version der Methode der Dynamischen Programmierung (DP-Algorithmus) an.

Bei der Methode der dynamischen Programmierung wird zunächst ein Gitter mit  $(n+1) \times (m+1)$  Punkten angesetzt, wobei im vorliegenden Ausführungsbeispiel  $n$  die Anzahl gesprochener Buchstaben und  $m$  die Anzahl erkannter Buchstaben ist. Die Zeilen des Gitters sind durch gesprochene Buchstaben und die Spalten des Gitters sind durch erkannte Buchstaben gekennzeichnet. Wie schon beim Gitter gemäß Fig. 2 sind die erste Zeile und die erste Spalte des Gitter nicht gekennzeichnet. Jedem durch ein Koordinatenpaar  $(i, j)$  mit  $i = 0, \dots, n$  und  $j = 0, \dots, m$  gekennzeichneten Gitterpunkt ist eine Wahrscheinlichkeit  $p_{ij}$  zugeordnet, die die Wahrscheinlichkeit dafür angibt, daß die Buchstabenfolge  $s_1, \dots, s_i$  eine Folge gesprochener Buchstaben (hier insbesondere ein Präfix eines eingegebenen Wortes, d.h. eine Folge aus mindestens einem Buchstaben, mit der das Wort beginnt) und daß  $r_1, \dots, r_j$  eine entsprechende zugehörige Folge erkannter Buchstaben ist. Der DP-Algorithmus ist eine Methode, um die Wahrscheinlichkeiten  $p_{ij}$  spaltenweise zu berechnen. Dabei wird die 0-te Spalte in jeder Zeile mit einer 1 initialisiert. Die Spalte  $i+1$  wird für  $i = 0, \dots, n-1$  aus der Spalte  $i$  bestimmt gemäß

$$\begin{aligned}
 p_{i+1,0} &= p_{i,0} P_R(\epsilon \mid s_{i+1}) P_S(s_{i+1} \mid s_1, \dots, s_i) \quad \text{und} \\
 p_{i+1,j+1} &= p_{i+1,j} P_R(r_{j+1}, \epsilon) \\
 &\quad + p_{i,j} P_R(r_{j+1} \mid s_{i+1}) P_S(s_{i+1} \mid s_1, \dots, s_i) \\
 &\quad + p_{i,j+1} P_R(\epsilon \mid s_{i+1}) P_S(s_{i+1} \mid s_1, \dots, s_i)
 \end{aligned}$$

für  $j = 0, \dots, m-1$ .

Hieraus ergibt sich durch Vergleich mit der oben angegebenen (und eine Produkt- und Summenbildung enthaltenden) Formel für  $P_R(r \mid s)$  die gesuchte Funktion  $f(s)$  gemäß

$$f(s) = p_{n,m} P_S(\$ \mid s).$$

Beginnen zwei Buchstabenfolgen  $\underline{s}_1$  und  $\underline{s}_2$  mit der gleichen Buchstabenfolge  $\underline{s}$  der Länge  $n$ , sind die ersten  $n$  Spalten des dem DP-Algorithmus zugrundegelegten Gitters identisch. Um dementsprechende redundante Berechnungen zu vermeiden, wird die folgende Modifikation vorgeschlagen: Die Spalten des DP-Gitters (eines dem DP-Algorithmus zugrundeliegenden Gitters) werden als Knoten eines Baumes definiert. Jeder Baumpfad entspricht nun

einem DP-Gitter und Baumpfade mit einem identischen Anfangssegment entsprechen zwei DP-Gittern für die Buchstabenfolgen  $s_1$  und  $s_2$  mit gleicher Anfangsbuchstabenfolge (anders ausgedrückt: gleichem Präfix). Fig. 3 verdeutlicht diesen Ansatz und zeigt die dem Beispiel gemäß Fig. 1 entsprechende Baumstruktur. Im dargestellten Beispiel wurden zwei Buchstaben erkannt, so daß jedem Baumknoten drei DP-Gitterknoten (entsprechend einer DP-Gitterspalte) zugeordnet sind.

Im folgenden wird nun ein Ansatz beschrieben, der zeigt, daß nicht alle Knoten einer solchen Baumstruktur bewertet werden müssen, um das Maximum der Funktion  $f(s)$  zu erhalten, und zwar unter Zuhilfenahme des sogenannten A'-Algorithmus.

Die Baumknoten werden nachstehend mit  $t^{(1)}, t^{(2)}, \dots$  bezeichnet. Der  $j$ -te Eintrag ( $j = 0, \dots, m$ ) in der Gitterspalte, die dem Knoten  $t^{(k)}$  zugeordnet ist, sei  $t_j^{(k)}$ . Weiterhin sei

$$t_s^{(k)} = t_m^{(k)} P_s(s \mid \underline{s}),$$

wobei die Buchstabenfolge  $\underline{s}$  die auf dem Pfad zum Knoten  $t^{(k)}$  liegende Buchstabenfolge ist. Nun kann das Problem, die Folge gesprochener Buchstaben mit der größten Wahrscheinlichkeit zu finden, in einer abgewandelten Form formuliert werden, und zwar als Suche nach dem Baumknoten  $t^{(k)}$ , für den der Wert  $t_s^{(k)}$  maximal ist.

Nach der Bewertung eines Baumknotens  $t^{(k)}$  wird ein oberer Grenzwert  $\tilde{t}^{(k)}$  geschätzt durch

$$\tilde{t}^{(k)} \geq \max \{ t_s^{(1)} \mid t^{(1)} \text{ ist Nachfolgeknoten von } t^{(k)} \}.$$

Nach der Bewertung zweier Baumknoten  $t^{(k)}$  und  $t^{(k')}$  und bei Vorliegen der Bedingung

$$\tilde{t}^{(k)} < t_s^{(k')}$$

weiß man bereits, daß kein Nachfolgebaumknoten des Baumknotens  $t^{(k)}$  ein optimaler Baumknoten sein kann. Eine Bewertung solcher Nachfolgebaumknoten erübrigt sich somit und wird nicht durchgeführt.

Zur Berechnung des Grenzwert  $\tilde{t}^{(k)}$  wird der sogenannte A\*-Algorithmus angewendet.

Die hier wesentlichen Iterationsschritte des an sich bekannten A\*-Algorithmus (siehe beispielsweise E.G. Schukat-Talamazzini, "Automatische Spracherkennung", Vieweg-Verlag, 1995, Kapitel 8.2.1) sind:

- (1) Initialisierung:  
Bewertung des Baumwurzelknotens.
- 10 (2) Iteration:  
E sei ein Satz schon bewerteter Knoten.  
Es gelte:  $p = \max \{ t_s \mid t \in E \}$ .  
Es gelte:  $\bar{p} = \max \{ \tilde{t} \mid t \in E \}$ .
- 15 (3) Überprüfung, ob das Beendigungskriterium erfüllt ist:  
Für  $p \geq \bar{p}$  : Ende des Algorithmus (keine weiteren Iterationsschritte notwendig).  
Der optimale Baumknoten ist der Knoten  $t \in E$ , für den  $t_s$  maximal ist.
- 20 (4) Expansions des Baums:  
Es wird ein bisher nicht expandierter Baumknoten  $t \in E$  ausgewählt und expandiert, was eine Bewertung aller seiner Tochterknoten beinhaltet. Danach wird nun der Algorithmus mit Schritt (2) fortgesetzt.
- 25 Zu beachten ist, daß in Schritt (4) grundsätzlich Freiheit in der Auswahl eines Knotens  $t \in E$  besteht. Um eine größtmögliche Effizienz des Algorithmus zu gewährleisten, ist es allerdings anzustreben, an dieser Stelle den Baumknoten zu wählen, der die größte Wahrscheinlichkeit besitzt, Teil des Pfades zum optimalen Baumknoten zu sein. Demgemäß wird hier derjenige Baumknoten  $t \in E$  gewählt, für den das Maximum  $\max \{ t_s \}$  maximal
- 30 ist, d.h. es wird der Baumknoten  $t \in E$  gewählt, der den meistwahrscheinlichen bereits bewerteten Gitterpunkt aufweist.



Nun soll näher darauf eingegangen werden, wie der Wert für  $\tilde{\tau}^{(k)}$  zu ermitteln ist. Grundsätzlich bestehen viele Möglichkeiten, diesen Wert zu bestimmen. Als eine vorteilhafte Möglichkeit der Bestimmung von  $\tilde{\tau}^{(k)}$  - bei der der Rechenaufwand gering gehalten wird und überflüssige Iterationsschritte vermieden werden - wird folgendes vorgeschlagen:

5

Es sei

$$\hat{j} = \arg \max_{j=0, \dots, m} \{t_j^{(k)}\}$$

und

10

$$c_j = \max \{P_R(r_j | s) \mid s \in \Sigma\}$$

für  $j = \hat{j} + 1, \dots, m$ . Der gesuchte Wert von  $\tilde{\tau}^{(k)}$  ergibt sich dann zu

$$\tilde{\tau}^{(k)} = \max \left\{ \begin{array}{l} t_m^{(k)}, \\ t_{m-1}^{(k)} c_m, \\ t_{m-2}^{(k)} c_m c_{m-1}, \\ \vdots \\ t_{\hat{j}}^{(k)} c_m c_{m-1} \dots c_{\hat{j}+1} \end{array} \right\}.$$

15

Die Berechnung dieses Ausdrucks für  $\tilde{\tau}^{(k)}$  ist mit wenig zusätzlichem Rechenaufwand verbunden, denn die Produkte  $c_m, c_m c_{m-1}, \dots$  können vorab berechnet werden und der Minimum-Index  $\hat{j}$  wird ohnehin in Schritt (4) des  $A^*$ -Algorithmus bestimmt.

- 20 Eine weitere Ausführungsvariante ergibt sich, wenn der  $A^*$ -Algorithmus beim erstmaligen Erfüllen der Bedingung im Schritt (3) nicht gestoppt, sondern weitere Algorithmus-Schleifen durchlaufen und weitere suboptimale Baumpfade ermittelt werden. Dann wird entsprechend N-1 weiteren durchlaufenen Schleifen statt einer einzigen Hypothese für die Folge gesprochener Buchstaben eine Liste N bester Hypothesen ausgegeben, und zwar
- 25 diejenigen, die am wahrscheinlichsten die Folge der gesprochenen Buchstaben wiedergeben.

Der oben beschriebene Algorithmus garantiert das Finden des optimalen Baumknotens und damit der optimalen Schätzung der eingegebenen Buchstabenfolge  $s$ ; er ist allerdings rechenzeitintensiv und benötigt viel Speicherplatz. Im folgenden soll erläutert werden, wie die Rechenzeit und der Speicherplatzbedarf verringert werden können. Bei dem entsprechend modifizierten A\*-Algorithmus werden lediglich die offenen Baumknoten gespeichert, d.h. die Baumknoten, die schon bewertet, aber noch nicht expandiert wurden. Nach der Expansion eines Baumknotens wird dieser im Speicher gelöscht. Die maximale Anzahl zu speichernder offener Baumknoten wird a priori vorgegeben. Liegt die Anzahl offener Baumknoten über dieser vorgegebenen maximalen Anzahl, ist demgemäß zu ermitteln, welche dieser offenen Baumknoten bei der nachfolgenden Berechnung außer Betracht bleiben können (sogenanntes "pruning"), wobei diese Knoten nicht zum optimalen Baumpfad gehören dürfen, da ansonsten der A\*-Algorithmus ein falsches Ergebnis liefern würde. Es stellt sich also hier das Problem, diejenigen Baumknoten zu finden, die mit der größten Wahrscheinlichkeit nicht Teil des optimalen Baumpfades sind. Zur Lösung dieses Problems wird ein einfacher heuristischer Ansatz gewählt. Dabei werden diejenigen offenen Baumknoten vorrangig als außer Betracht zu bleibende Baumknoten gewählt, die am nächsten zur Baumwurzel liegen. Dies bedeutet, daß Suchpfade, die schon zu Beginn eine kleine Wahrscheinlichkeit aufweisen, diejenigen Baumpfade sind, die vorrangig als nicht mehr weiterzuverfolgende Pfade eingestuft werden.

Die beschriebene "pruning"-Strategie kann effizient insbesondere dadurch implementiert werden, daß die offenen Knoten nicht in einer gemeinsamen Halde ("heap") gespeichert werden, sondern für jede Pfadlänge einer Halde vorgesehen ist und die offenen Knoten in der jeweils zugehörigen Halde gespeichert werden. Im Fall einer Überschreitung der zulässigen Anzahl offener Knoten (s.o.) wird bei dieser Form der Implementierung diejenige Halde, die den kürzesten Baumpfad repräsentiert, gelöscht. Die hierfür erforderliche Zeitspanne ist nahezu konstant.

Fig. 4 zeigt ein Blockschaltbild eines Spracherkennungssystems 1 zur Erkennung eingegebener buchstabierter sprachlicher Äußerungen  $s$ , daß gemäß den obigen Ausführungen zur erfindungsgemäßen Buchstaben-Spracherkennung arbeitet. Ein Block 2 kennzeichnet eine

Spracherkennungseinheit, die basierend auf akustischen Modellen - wie üblich werden HMM (Hidden Markov Modelle) verwendet - ein Erkennungsergebnis  $\underline{r}$  (Folge von Buchstaben) liefert, wobei eine Buchstabengrammatik, die Wahrscheinlichkeiten für das Auftreten verschiedener möglicher Buchstabenkombinationen angibt, durch die Sprach-  
 5 erkennungseinheit 2 nicht verwendet wird. Das Erkennungsergebnis  $\underline{r}$  wird einer Nach-  
 verarbeitungseinheit 3 zugeführt, die basierend auf durch einen Block 4 dargestellten  
 statistischen Modellen für Buchstabenfolgen  $P_s(s)$  und auf mit einem Block 5 dargestellten  
 statistischen Modellen  $P_r(\underline{r} | s)$  für den Spracherkenner wie oben beschrieben die entspre-  
 chende Funktion  $f(s)$  maximiert (Block 6) und hieraus ein auszugebendes Erkennungs-  
 10 ergebnis  $R_s$  ableitet. Das Erkennungsergebnis  $R_s$  ist entweder eine Schätzung der eingege-  
 benen Buchstabenfolge  $s$  oder eine Liste  $N$  bester Schätzungen der eingegebenen Buch-  
 stabenfolge  $s$  mit den größten Wahrscheinlichkeiten, die richtige Schätzung zu sein.

Das in Fig. 5 dargestellte Blockschaltbild zeigt ein System mit Sprachsteuerung - hier  
 15 vorzugsweise ein Navigationssystem für Landkraftfahrzeuge -, das sowohl einen Buchsta-  
 benspracherkenner 1 gemäß Fig. 4 als auch einen Wort-Spracherkenner 7 zur Erkennung  
 eingegebener Worte  $w$  aufweist. Zur Anwendung der Erfindung kommen aber prinzipiell alle  
 sprachgesteuerten Systeme mit Funktionseinheiten zur Erkennung buchstabierter Sprach-  
 äusserungen in Betracht. Das vom Buchstaben-Spracherkenner 1 gelieferte Erkennungs-  
 20 ergebnis  $R_s$  wird zur Einschränkung des Vokabulars des Wort-Spracherkenners 7, d.h. zur  
 Begrenzung der als Wort-Erkennungsergebnis  $R_w$  in Frage kommenden Worte, verwendet,  
 was zu einer robusteren Wort-Spracherkennung führt. Bei einer bestimmten Anfangsbuch-  
 stabenfolge oder  $N$  bester Anfangsbuchstabenfolgen als Erkennungsergebnis  $R_s$  wird das  
 Vokabular des Wort-Spracherkenners 7 auf die Worte mit diesen Anfangsbuchstabenfolgen  
 25 eingeschränkt. Das Erkennungsergebnis  $R_w$  wird zur Systemsteuerung verwendet, wobei  
 die gesteuerten Systemfunktionseinheiten durch einen Block 8 zusammengefaßt sind. Bei  
 Navigationssystemen stellt das Erkennungsergebnis beispielsweise einen Ortsnamen dar,  
 dessen Eingabe das Navigationssystem zur Ermittlung einer dorthin führenden Fahrstrecke  
 veranlaßt.

Vorliegend werden der Spracherkennungsblock 2 und die Nachverarbeitungseinheit 3 mittels unterschiedlicher Hardware-Komponenten umgesetzt - der Spracherkennungsblock mittel eines für Spracherkennungsaufgaben angepaßten digitalen Signalprozessors und die Nachverarbeitungseinheit 3 mittels eines auch zur Realisierung anderer durch Block 8 zusammengefaßter Systemfunktionen dienenden Controllers. Dies hat den Vorteil, daß der Signalprozessor eine geringere Rechenkapazität und eine geringere Speicherkapazität aufweisen kann, da zur Erkennung eingegebener buchstabierter Sprachäußerungen sonst für Navigationsprozeduren vorgesehene Systemressourcen mitbenutzt werden.

10

15

## PATENTANSPRÜCHE

1. Verfahren zur Erkennung einer in buchstabierter Form vorliegenden Sprachäußerungseingabe (s) mit einer ersten Verarbeitungsstufe, in der mittels einer auf Hidden Markov Modellen basierenden Buchstaben-Spracherkennungseinheit (2) eine korrespondierende Buchstabenfolge (r) geschätzt wird, und mit einer zweiten Verarbeitungsstufe (3), in der
- 5 unter Verwendung eines statistischen Buchstabenfolgemodells (4) und eines statistischen Modells (5) für die Spracherkennungseinheit (2) das von der ersten Verarbeitungsstufe gelieferte Schätzergebnis (r) nachverarbeitet wird, wobei bei der Nachverarbeitung die Methode der dynamischen Programmierung eingesetzt wird,  
dadurch gekennzeichnet,
- 10 daß die der dynamischen Programmierung zugrundeliegende Gitterstruktur, deren Knotenpunkte zur Zuordnung zu akkumulierten Wahrscheinlichkeitswerten vorgesehen sind, in eine Baumstruktur umgesetzt wird und  
daß bei der Suche nach einem optimalen Baumpfad der A\*-Algorithmus eingesetzt wird.
- 15 2. Verfahren nach Anspruch 1,  
dadurch gekennzeichnet,  
daß suboptimale Baumpfade entsprechend N bester Schätzungen für die Sprachäußerungseingabe mit  $N > 1$  ermittelt werden.
- 20 3. Verfahren nach Anspruch 1 oder 2,  
dadurch gekennzeichnet,  
daß bei der Suche nach einem optimalen Baumpfad diejenigen Baumpfade, die schon zu Beginn der Suche eine gegenüber anderen Baumpfaden eine kleine Wahrscheinlichkeit aufweisen, vorrangig nicht mehr weiterverfolgt werden.

4. Verfahren nach einem der Ansprüche 1 bis 3,

dadurch gekennzeichnet,

daß die erste Verarbeitungsstufe mittels eines ersten ICs und die zweite Verarbeitungsstufe mittels eines zweiten ICs durchgeführt wird.

5

5. Verfahren zur Systemsteuerung mittels Sprachsignalen (w, s), bei dem

- eine Eingabe eines als Steuersignal dienenden ganzen Wortes (w) und eine Eingabe wenigstens eines Teils dieses Wortes in buchstabierter Form (s) vorgesehen ist,
- eine Wort-Spracherkennung (7) zur Erkennung des eingegebenen ganzen Wortes (w) vorgesehen ist,

10

- eine Buchstaben-Spracherkennung (1), insbesondere nach einem der Ansprüche 1 bis 4, zur Erkennung des eingegebenen buchstabierten Teils (s) des ganzen Wortes (w) vorgesehen ist und

15

- bei dem mit Hilfe des Erkennungsergebnisses (s) der Buchstaben-Spracherkennung (1) eine Einschränkung eines der Wort-Spracherkennung (7) zugeordneten Vokabulars durchgeführt wird.

20

6. Sprachgesteuertes elektrisches Gerät, insbesondere Navigationssystem für Landkraftfahrzeuge, mit Komponenten (1, 7, 8) zur Durchführung eines Verfahrens nach einem der Ansprüche 1 bis 5.

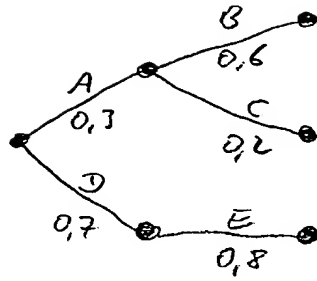


Fig. 1

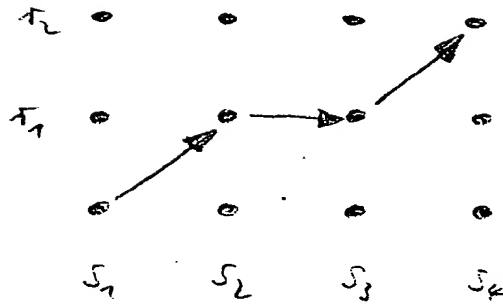


Fig. 2

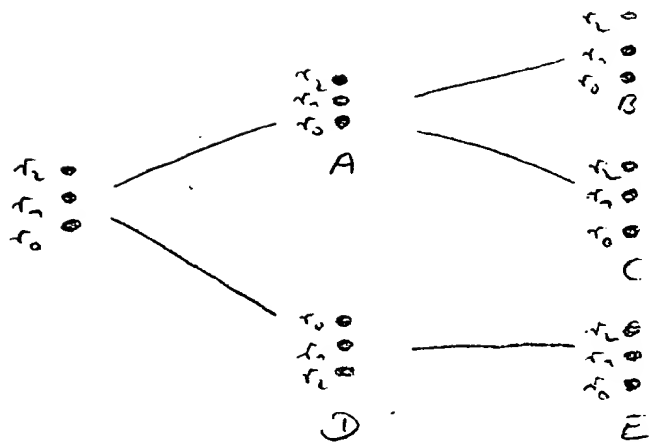


Fig. 3

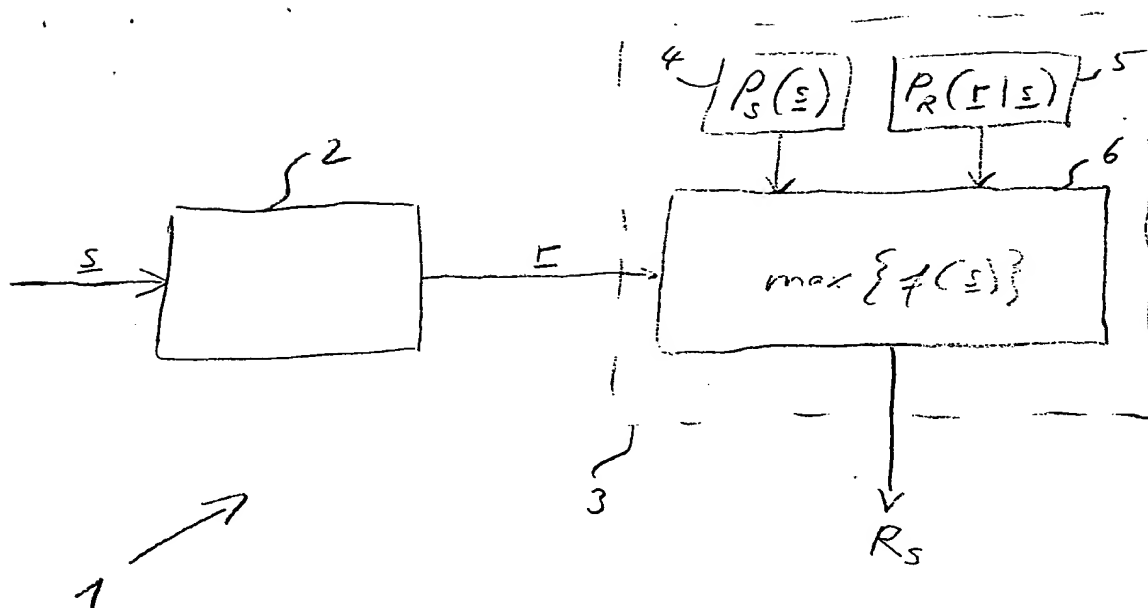


Fig. 4

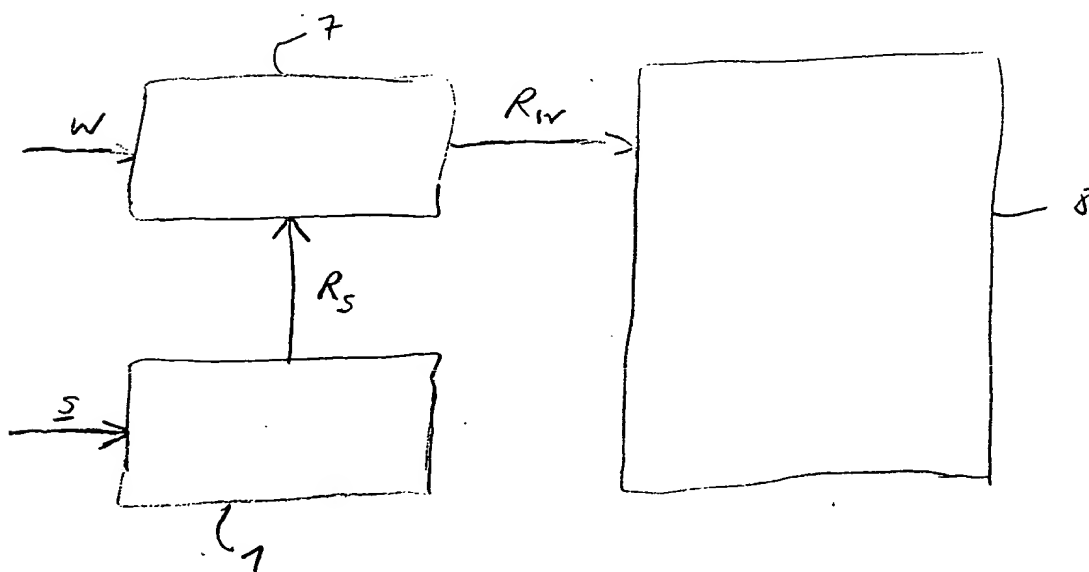


Fig. 5